Theoretical Computer Science 18 (1982) 105–111
North-Holland Publishing Company

NOTE

THE COMPLEXITY OF PRESBURGER ARITHMETIC WITH BOUNDED QUANTIFIER ALTERNATION DEPTH*

Martin FÜRER

Department of Computer Science, University of Edinburgh, Edinburgh EH9 3JZ, United Kingdom

Communicated by A. Meyer
Received February 1981
Revised September 1981

Abstract. It is shown how the method of Fischer and Rabin can be extended to get good lower bounds for Presburger arithmetic with a bounded number of quantifier alternations. In this case, the complexity is one exponential lower than in the unbounded case. This situation is typical for first order theories.

1. Introduction

Almost all lower bounds for the complexity of the decision problem for mathematical theories rely on the fact that certain formulas with many quantifier alternations are hard to decide. Mathematicians realized a long time ago that formulas with more than a few quantifier alternations are no longer understandable. Therefore it is justified to pay special attention to formulas with few quantifier alternations and to investigate the complexity of the decision problem not only as a function of the length of formulas, but as a function of length and quantifier alternation depth, and in particular to investigate this complexity for the formulas of constant depth.

Definition. For $\rho \in \{\sigma, \delta\}$ the depth $D_\rho(F)$ of a formula F is defined by:

$$D_\rho(F) = 0 \text{ if } F \text{ is quantifier-free,}$$

$$D_\rho(F \vee G) = D_\rho(F \wedge G) = \max(D_\rho(F), D_\rho(G)),$$

$$D_\sigma(\neg F) = D_\delta(F), \qquad D_\delta(\neg F) = D_\sigma(F)$$

$$D_\rho(F \to G) = D_\rho(\neg F \vee G)$$

$$D_\rho(F \leftrightarrow G) = D_\rho((\neg F \vee G) \wedge (F \vee \neg G)),$$

* This work was supported by the Science Research Council under grant GR/A 80334 and by the Swiss National Foundation.

$$D_\sigma(\exists xF) = \max(D_\sigma(F), 1),$$

$$D_\delta(\exists xF) = D_\sigma(\exists xF) + 1,$$

$$D_\rho(\forall xF) = D_\rho(\neg\exists x\neg F).$$

The *quantifier alternation depth* of a formula F is $\min(D_\sigma(F), D_\delta(F))$.

Presburger arithmetic (PA) is the theory of nonnegative integers $\mathbb{N} = \{0, 1, 2, \ldots\}$ with 0, 1 and +.

Let PA(m) be the set of formulas of PA whose quantifier alternation depth is at most m. Reddy and Loveland [9] have shown that PA(m) is accepted by a deterministic Turing machine in space $2^{cn^{m+4}}$. (They have proved this result for formulas in prenex normal form, but the extension to PA(m) is obvious.) The constants c and d are always independent of m and n. (m can be considered as a constant or a function of the input length n.) We assume the reader is familiar with the lower bound theorem of Fischer and Rabin [4] for PA, and we show how this theorem can easily be extended to get good lower bounds for PA(m) too.

2. Lower bound

Let L be the set of formulas in first order logic with 0, 1, + and =. The extended version of the crucial Theorem 8 in [4] is:

Theorem 1. *There exists a constant $c > 0$ so that for all non-negative integers m and k there is a formula $M_k^m(x, y, z)$ of L, such that for real numbers, A, B, C,*

$$M_k^m(A, B, C) \text{ is true} \leftrightarrow A \in \mathbb{N} \wedge A \leq 2^{k^m} \wedge AB = C.$$

Also $M_k^m(x, y, z)$ is a formula with quantifier alternation depth $\max(0, 2m-1)$ *and length at most* $c(mk \log k + 1)$, *and $M_k^m(x, y, z)$ is Turing machine computable from m and k in time polynomial in m and k.*

Proof. The construction of the formulas $M_k^m(x, y, z)$ will be inductive on m. For $m = 0$ we have $2^{k^0} = 2$ and we define $M_k^0(x, y, z)$ as

$$(x = 0 \wedge z = 0) \vee (x = 1 \wedge z = y) \vee (x = 1 + 1 \wedge z = y + y).$$

To get M_k^{m+1} from M_k^m, we observe that for $q \in \mathbb{N} - \{0\}$ and $k > 0$,

$$x \in \mathbb{N} \quad \text{and} \quad x \leq q^k$$

is equivalent to

$$\text{there exist } x_0, \ldots, x_{k-1} \in \mathbb{N} \text{ with } x_0 \leq q, x_1 < q, x_2 < q, \ldots, x_{k-1} < q$$

$$\text{and } x = \sum_{i=0}^{k-1} x_i q^i$$

and this is equivalent to

$$\text{there exist } u, x_0, \ldots, x_{k-1} \in \mathbb{N} \text{ with } u \leq q,\ x_0 \leq q,\ x_1 < q,$$
$$x_2 < q, \ldots, x_{k-1} < q \text{ and } x = \sum_{i=0}^{k-1} x_i u^i.$$

If $x = \sum_{i=0}^{k-1} x_i u^i$, then $xy = \sum_{i=0}^{k-1} x_i y u^i$ and by Horner

$$x = x_0 + u(x_1 + u(\cdots + u(x_{k-2} + ux_{k-1})\cdots))$$

and

$$xy = x_0 y + u(x_1 y + u(\cdots + u(x_{k-2}y + ux_{k-1}y)\cdots)).$$

Hence, for $k > 0$, $M_k^{m+1}(x, y, z)$ is equivalent to

$$\exists u, x_0, \ldots, x_{k-1}, z_0, \ldots, z_{k-1}, v_0, \ldots, v_{k-1}, w_0, \ldots, w_{k-1}$$
$$\bigwedge_{i=1}^{k-1} x_i \neq u \wedge \bigwedge_{i=1}^{k-1} M_k^m(u, x_i + v_i, v_{i-1}) \wedge v_{k-1} = 0 \wedge x = x_0 + v_0$$
$$\wedge \bigwedge_{i=0}^{k-1} M_k^m(x_i, y, z_i) \wedge \bigwedge_{i=1}^{k-1} M_k^m(u, z_i + w_i, w_{i-1}) \wedge w_{k-1} = 0$$
$$\wedge z = z_0 + w_0.$$

If we replace $x_i \neq u$ by a formula expressing $x_i < u$, then we get a formula which is more easily understandable.

We define M_k^1 exactly like this, but for $m > 0$, M_k^{m+1} is constructed from this formula using the abbreviation method [4]:

$$\bigwedge_{i=1}^{p} M(r_i, s_i, t_i)$$

is replaced by

$$\forall x \forall y \forall z \left[\left(\bigvee_{i=1}^{p} x = r_1 \wedge y = s_i \wedge z = t_i\right) \rightarrow M(x, y, z)\right].$$

And as in [4], the same variable names are used several times, so that only $4k+7$ different variable names appear in $M_k^m(x, y, z)$. (Actually variable renaming is not necessary when $m = O(k)$ as in Theorem 2.)

Then there is a constant c with

$$|M_k^{m+1}(x, y, z)| \leq |M_k^m(x, y, z)| + ck \log k$$

and therefore

$$|M_k^m(x, y, z)| \leq c(mk \log k + 1).$$

The quantifier alternation depth of $M_k^m(x, y, z)$ is $\max(2m-1, 0)$. It is obvious that $M_k^m(x, y, z)$ is computable by a Turing machine in time polynomial in m and k (even within space $O(\log m + \log k)$ on the work tapes).

The method we use to get lower complexity bounds from a formula for multiplication is not new. Therefore we present only an outline.

First we use modular arithmetic and the prime number theorem, as in [4], to define multiplication up to $2^{2^{(n/m)^{cm}}}$ in Presburger arithmetic by a formula of length n and quantifier alternation depth m.

To do this, we define $g(n)$ to be the least common multiple of all positive integers less than or equal to n. Then $g(n) = e^{\psi(n)}$, where ψ is Tchebychef's function. Using this definition of g, it is easy to write a formula $G_k^m(x)$ which is true precisely for $x = g(2^{k^m})$. (Note that y is a non-negative integer $\leq 2^{k^m}$ iff $M_k^m(y, 0, 0)$ is true.)

Let $f_1(n) \sim f_2(n)$ denote asymptotic equivalence of f_1 and f_2, i.e. $\lim_{n \to \infty} f_1(n)/f_2(n) = 1$.

Then the prime number theorem (see e.g. [7]) is equivalent to $\psi(n) \sim n$. (In fact one proves $\psi(n) \sim n$ in order to prove the prime number theorem.) And the Chinese remainder theorem implies that for all non-negative integers $x, y, z < g(n)$, we have

$$xy = z \quad \text{iff}$$

$$x_i y_i = z_i \quad \text{for all positive integers } i \leq n$$

$$\text{and all } x_i, y_i, z_i < i \quad \text{with}$$

$$x \equiv x_i,\ y \equiv y_i \text{ and } z \equiv z_i \mod i.$$

Using the formulas $M_k^m(x, y, z)$ and $G_k^m(x)$ to express these congruences and equalities, we can easily define a formula $\mathrm{Prod}_k^m(x, y, z)$ which is true in $\mathbb{N}$ iff $xy = z$ and $x, y, z < g(2^{k^m})$. The length of $\mathrm{Prod}_k^m(x, y, z)$ is only $c(mk \log k + 1)$ and its quantifier alternation depth is $2m + d$. We have already seen that for all constants $c < e$ and for k^m sufficiently big $g(2^{k^m}) > c^{2^{k^m}}$.

Now when multiplication is defined for a large initial segment of $\mathbb{N}$, we can—if we like—depart from the paper of Fischer and Rabin and follow the more elegant method used by Heintz [8] and A. R. Meyer (see [5]): we first define the concatenation of words and then use it to describe computations of Turing machines. All this increases the quantifier alternation depth of the formula only by a small additive constant.

The most elegant method to code words over a finite alphabet into non-negative integers is due to Smullyan [10]. The number $A = \sum_{i=0}^{n} a_i 2^i$ with $a_i \in \{1, 2\}$ (not $\{0, 1\}$!) codes the word $W(A) = a_n a_{n-1} \ldots a_0$. This defines a one-to-one enumeration of the words over $\{1, 2\}$.

We first define a formula $\mathrm{Pot}_k^m(y, u)$ saying that $y < g(2^{k^m})$ and u is the biggest power of 2 less than or equal to y. Then the following formula says that $W(z)$ is the concatenation of $W(x)$ with $W(y)$ and $z < g(2^{k^m})$ (hence the length of $W(z)$ is bounded by $\log g(2^{k^m}) \sim 2^{k^m}$):

$$\exists u\, \exists v\ \mathrm{Pot}_k^m(y+1, u) \wedge \mathrm{Prod}_k^m(x, u, v) \wedge v + y = z.$$

Having concatenation available it is no problem (but a bit lengthy) to describe computations. Here 'computation' means a suitable encoding of a sequence of subsequent configurations of a Turing machine.

We can describe computations of length 2^{k^m} by formulas of length $c_1(mk \log k + 1)$ and quantifier alternation depth $2m + c_2$. Hence, there are constants c and d such that for all $m, n > d$ we can describe computations of length $< 2^{(n/m)^{cm}}$ by formulas of length n and depth m. This implies:

Theorem 2. *There is a constant $c > 0$ such that no $2^{(n/m)^{[cm]}}$ time-bounded non-deterministic Turing machine can decide* PA(m).

This result holds also when m is a function of the input length n.

3. The Berman complexity

Both the time and the space complexity measure seem to be not quite adequate for mathematical theories like Presburger arithmetic. Berman [1] has shown that the simultaneous bounds for time and alternation depth of an alternating Turing machine is a better complexity measure.

Directly applying the method of [1] to the above non-deterministic lower time bound for PA(m) and to the upper space bound of [9] for PA(m), we get (as for PA) also pretty tight bounds for PA(m):

Theorem 3. *There is a constant $c > 0$ such that for all $m, m' \in \mathbb{N}$ (with $m' < m$) $2^{(n/m)^{[c(m-m')]}}$ is a lower bound for the time complexity of alternating Turing machines deciding* PA(m) *with at most m' alternations.*

Remark. Theorem 3 can be improved, if we say in a formula, "there is an accepting computation tree", instead of simulating the alternations of the alternating Turing machine by quantifier-alternations of the formula.

Theorem 4. *There is a constant c such that for all $m \in \mathbb{N}$ there is a $2^{cn^{m+4}}$ time-bounded alternating Turing machine which decides* PA(m) *with m alternations.*

4. Other theories

The method of this paper can be applied to other mathematical theories. The exponential complexity jump from PA(m) (for constant m) to PA is typical for those mathematical theories whose decision complexity is at least non-deterministic exponential time.

If the complexity is also not more than exponential space, such as for the theory of reals with addition [3], then good lower bounds for bounded and for unbounded quantifier alternation depth are usually much easier to obtain. It is then usually possible to define concatenation directly, without defining multiplication. We illustrate this method with one example.

4.1. *The theory of reals with addition*

We represent some reals (namely the non-negative integers) by words over a finite alphabet and define their concatenation in the theory of reals with addition. Instead of the theory of reals with addition, we could choose any other theory such that all its models are groups, and in one model an element of infinite order exists (see [8]).

As before, let the integer $A = \sum_{i=0}^{n} a_i 2^i$ with $a_i \in \{1, 2\}$ represent the word $W(A) = a_n \cdots a_0$. Hence the integer 0 represents the empty word. Now we define a formula $K_k^m(x, y, z)$, such that for all $A, B, C \in \mathbb{N}$, $K_k^m(A, B, C)$ is true iff $W(A)W(B) = W(C)$ (the concatenation of $W(A)$ with $W(B)$ is $W(C)$) and $B \leq 2^{k^m}$.

$K_k^0(x, y, z)$ is defined as

$$(y = 0 \wedge x = z) \vee (y = 1 \wedge x + x + 1 = z) \vee (y = 1 + 1 \wedge x + x + 1 + 1 = z)$$

and $K_k^{m+1}(x, y, z)$ is defined to be equivalent to

$$\begin{aligned} &\exists u_1, \ldots, u_k, y_0, \ldots, y_k, z_0, \ldots, z_k \\ &\quad \bigwedge_{i=1}^{k} K_k^m(y_{i-1}, u_i, y_i) \\ &\quad \wedge y_0 = 0 \wedge y_k = y \\ &\quad \wedge \bigwedge_{i=1}^{k} K_k^m(z_{i-1}, u_i, z_i) \wedge z_0 = x \wedge z_k = z. \end{aligned}$$

Using the abbreviation methods of Section 2, this formula can be transformed such that

$$|K_k^m(x, y, z)| \leq c(mk \log k + 1)$$

and the quantifier alternation depth of $K_k^m(x, y, z)$ is $\max(2m - 1, 0)$.

From this result it follows easily:

Theorem 5. *There is a constant $c > 0$ such that the validity of formulas with quantifier alternation depth $\leq m$ of the theory of reals with addition cannot be decided by an $(n/m)^{\lfloor cm \rfloor}$ time-bounded non-deterministic Turing machine.*

And naturally also the analogues to the Theorems 3 and 4 hold for the reals with addition.

5. Conclusions

For a big enough constant m, the complexity of PA(m) is one exponential lower than that of PA. This is typical for mathematical theories. For very small constants m, the complexity is even lower. PA(1) is (NP $\cup$ co-NP)-complete. This follows from the result of Borosh and Treybig [2] and von zur Gathen and Sieveking [6] that solvable linear diophantine inequality systems have a polynomially bounded solution. But for a constant m as low as about 6, we have an exponential nondeterministic lower time bound for PA(m). Hence, there is a very interesting complexity jump somewhere between $m = 1$ and $m = 10$.

Problem. Where does this complexity jump occur? In particular, for which $m \in \{1, \ldots, 10\}$ is PA(m) in NP $\cup$ co-NP or in POLYSPACE, and for which m is PA(m) not in NEXPTIME?

References

[1] L. Berman, The complexity of logical theories, *Theoret. Comput. Sci.* **11** (1980) 71–77.

[2] I. Borosh and L.B. Treybig, Bounds on positive integral solutions of linear diophantine equations, *Proc. AMS* **55** (1976) 299–304.

[3] J. Ferrante and C. Rackoff, A decision procedure for the first-order theory of real addition with order, *SIAM J. Comput.* **4** (1975) 69–76.

[4] M.J. Fischer and M.O. Rabin, Super-exponential complexity of Presburger arithmetic, *SIAM-AMS Proc.* **7** (1974) 27–41.

[5] K. Fleischmann, B. Mahr and D. Siefkes, Bounded concatenation theory as a uniform method for proving lower complexity bounds, in: R.O. Gandy and J.M.E. Hyland, Eds., *Logic Colloquium 76* (North-Holland, Amsterdam, 1977) 471–490.

[6] J. von zur Gathen and M. Sieveking, A bound on solutions of linear integer equalities and inequalities, *Proc. AMS* **72** (1978) 155–158.

[7] G. H. Hardy and E. M. Wright, *An Introduction to the Theory of Numbers* (Oxford University Press, Oxford, 1938).

[8] J. Heintz, Untere Schranken für die Komplexität logischer Entscheidungsprobleme, in: E. Specker and V. Strassen, Eds., *Komplexität von Entscheidungsproblemen*, Lecture Notes in Computer Science **43** (Springer, Berlin, 1976) 127–137.

[9] C.R. Reddy and D.W. Loveland, Presburger arithmetic with bounded quantifier alternation, *Proc. 10th Annual ACM Symposium on Theory of Computing* (1978) 320–325.

[10] R. Smullyan, *Theory of Formal Systems*, Annals of Mathematics Studies **47** (Princeton University Press, Princeton, 1961).